An XML Viewer for Tabular Forms for Use with Mechanical Documentation

Osamu INOUE †, Shun-ichi NAKAGAWA ††,
Tomokazu ARITA ††Takeo YAKU ††, Kensei TSUCHIDA†
Toyo Univ.† Nihon Univ.††

Contents

- Introduction
- Tabular Forms and Their Syntax
  - XML representation for Tabular Forms
- Parsing of a tabular forms
- Attribute Evaluation and XML viewer
  - Attribute rules for XML
- System Overview
- Conclusion

1. Introduction

- Background
- Our Purpose

Background

- Our Project
  - Implementation for a Programming Environment
  - Visualization for Diagram Structures
  - Formalization for diagram structures

Background (continued)

- Related projects for syntax-directed processing of graphs and diagrams.
  - APPLIGRAPH, IPSEN, DiaGen etc.
- Our project for a graph processing environment.
  - KEYAKI

Background (Continued)

- Formalism for Tabular Forms based on Attributed Graph
  - Formalism and Parsing for Tabular Forms based on an Attribute edNCE Graph Grammar

    (Arita et al. IFIP WCC ICSE2000)

Purpose

- To Generate XML source files for tabular forms from attributed graphs.
  - Representation of tabular forms by XML.
  - Definitions of Style sheets based on XSL.
  - Extension of an attribute graph grammar.
  - Development of tabular form processing system (HiformED).

2. Preliminaries

- Program Specification Form: Hiform
- edNCE Graph Grammr
- Marked Graph
- XML

Program Specification Forms (Hiform)

- An Example of a Hiform document (A1.General Document)

ProgramName :	General Document	A1 p
Subtitle :		
Library Code :	Version :	
Author :	Original Release :	
Approver :	Current Release :	
Key Words :	CR Code :	
Scope :		
Variants :		
Language :	Software Req. :	
Operation : Interactive Batch RealTime( )	Hardware Req. :	
References :		
Function :1. List and Explanation of InputData or Parameters 2. List and Explanation of OutputData or Result Values		
Example :		

edNCE Graph Grammr
[Rozenbarg 1997]

An edNCE graph grammar : $G = (\Sigma, \Delta, \Gamma, \Omega, P, S)$,

where

Σ : the alphabet of node labels,

$\Delta \subseteq \Sigma$: the alphabet of terminal node labels,

Γ : the alphabet of edge labels,

$\Omega \subseteq \Gamma$: the alphabet of final edge labels,

P : the finite set of productions,

$S \in \Sigma - \Delta$: the initial nonterminal.

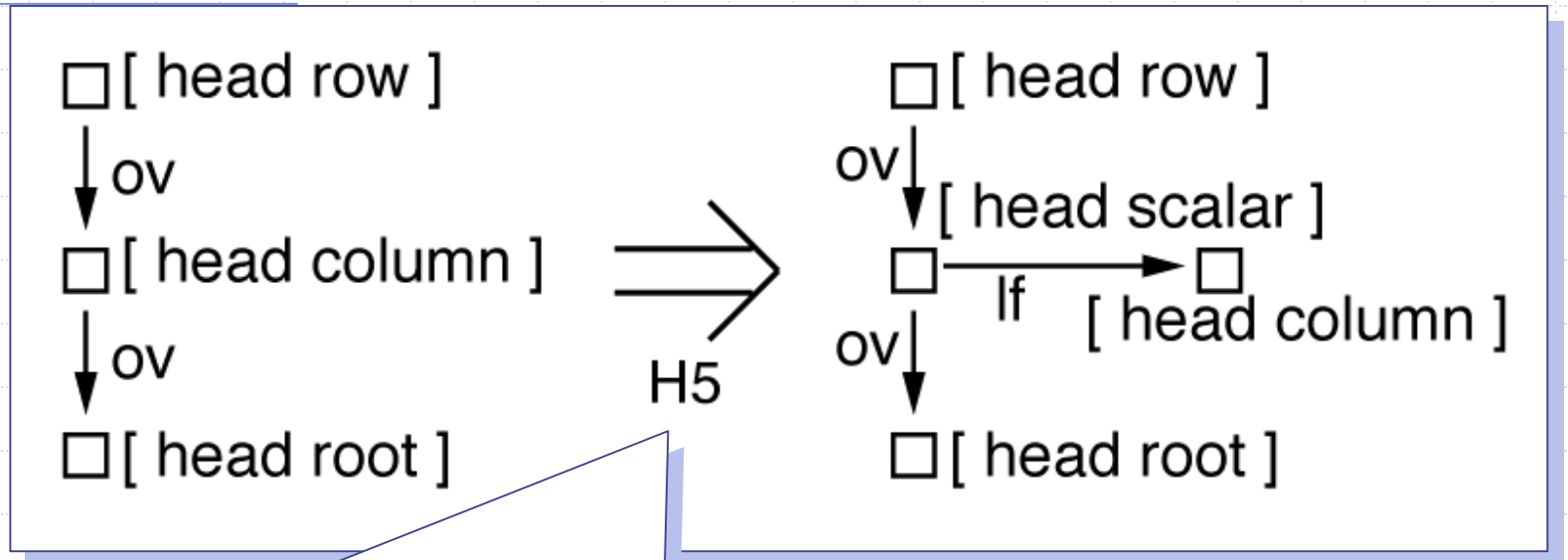
edNCE Graph Grammar (Continued)

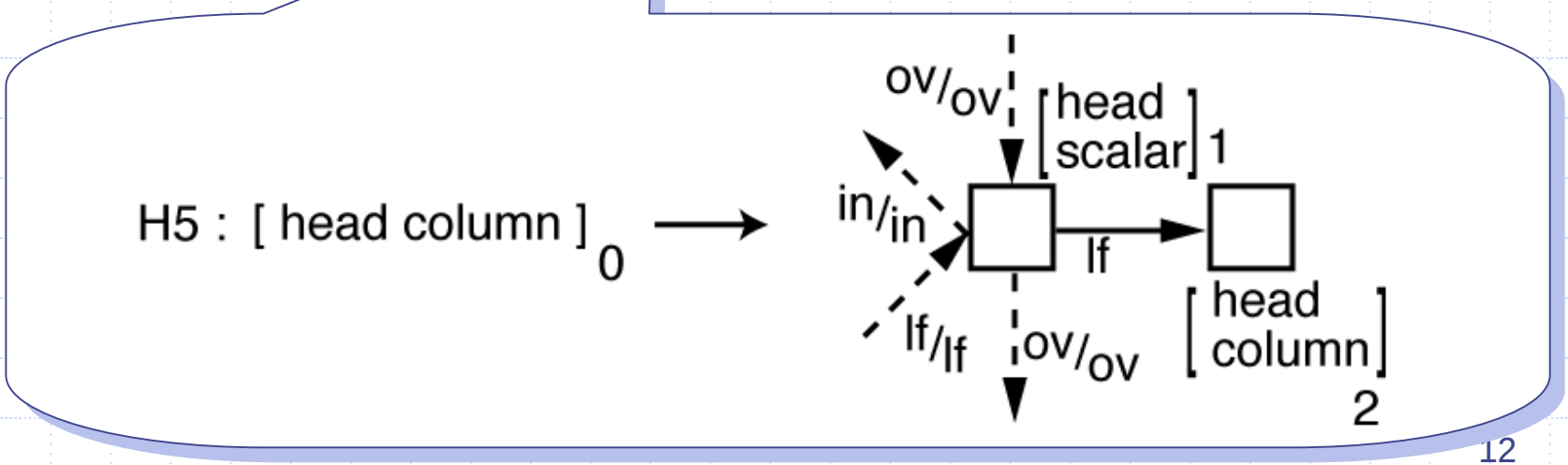
A production : $X \to (D, C)$

$X \in \Sigma - \Delta$,
D : a graph over the Σ and Γ,
C : the connection relation,
$C \subseteq \Sigma \times \Gamma \times \Gamma \times V_D \times \{in, out\}$
where V_D : a set of nodes on D.

□

An Example of Derivation by applying a production





An Attribute NCE Graph Grammar

[Arita et al, IASTED AI'01]

An attribute NCE graph grammar :
$AGG = \langle G, Att, F \rangle$ where

1. $G = (\Sigma, \Delta, \Gamma, \Omega, P, S)$:
   an underlying graph grammar of AGG.
2. $Att = \bigcup_{Y \in \Sigma} Att(Y)$,
   $(Att(Y) = Inh(Y) \cup Syn(Y).)$
3. $F = \bigcup_{p \in P} F_p$:
   the set of semantic rules of AGG.

An Attribute Graph Grammar for Tabular forms: (HNGG) [IASTED AI'01]

Hiform Nested tabular form Graph Grammar :

$HNGG = (G_N, Att_N, F_N)$,

where

$G_N = (\Sigma_N, \Delta_N, \Gamma_N, \Omega_N, P_N, S_N)$ s.t.

- Σ_N : node labels,
- $\Delta_N \subseteq \Sigma$: for items of program specifcations,
- $\Gamma_N = \{in, ov, lf\}$: for relations between items,
- $\Omega_N = \Gamma_N$
- P_N : the finite set of productions,
- $S_N = [struct]$

$Att_N = \{x, y, width, height\}$

F_N : used for drawing tabular forms.

An Attribute Graph Grammar for Tabular forms (Continued)

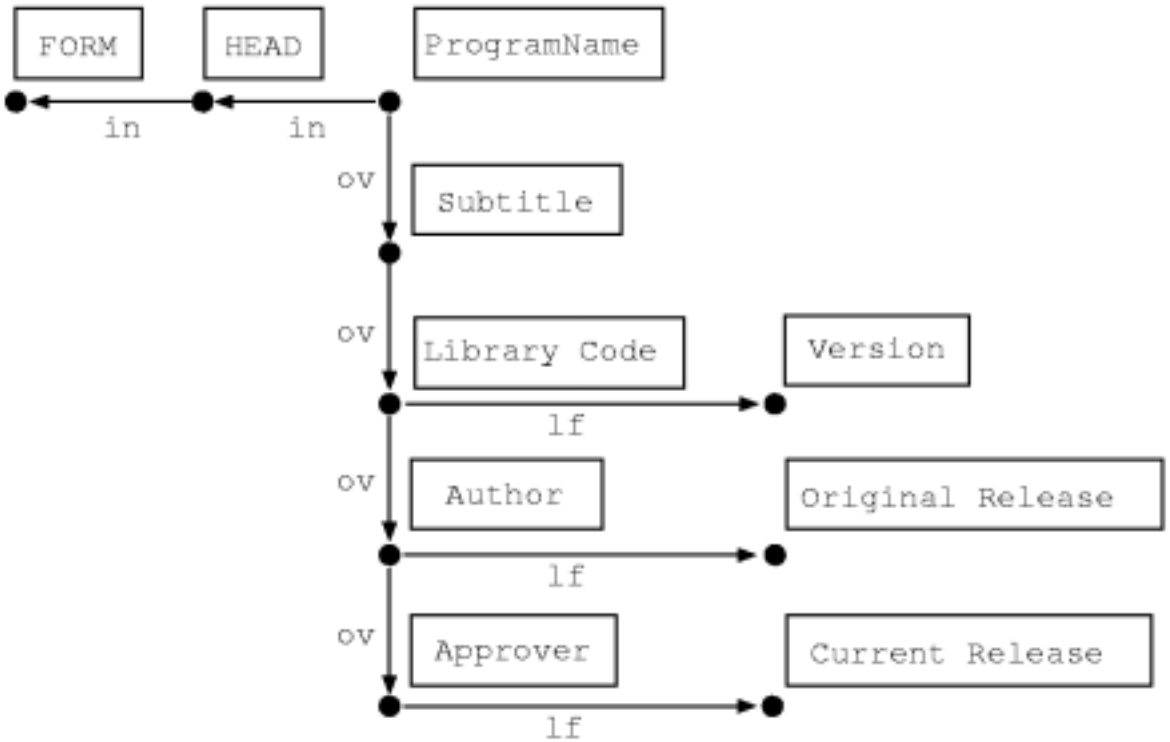
The Size of HNGG

Productions	280
Attribute rules	**1528**
Precedence relations	5376

In this paper, we extend attribute rules for generating XML code.

Marked Graph

- Hiform document and its corresponding marked graph
  - A marked graph T1 which represents tabular form



Marked Graph (Contunued)

- A tabular form for a marked graph T1

T1:

ProgramName :	
Subtitle :	
Library Code :	Version :
Author :	Original Release :
Approver :	Current Release :

XML(Extensible Markup Language)

- Meta Markup Language
- Describe a document structure simply.
- Web-based Language

- Markup Languages defined by XML:
  - MathML, SMIL, VML, SVG, XSL etc.

XSL(XML Stylesheet Language)[W3C 2001]

- Style sheet for XML
- Defined by XML
- Exchange XML file to other file,
  For example, XML to HTML

Representation of tabular forms by XML

- To represent of tabular forms by XML
- Definitions of display for XML documents by XSL files

Representation of tabular forms by XML (Contimued)

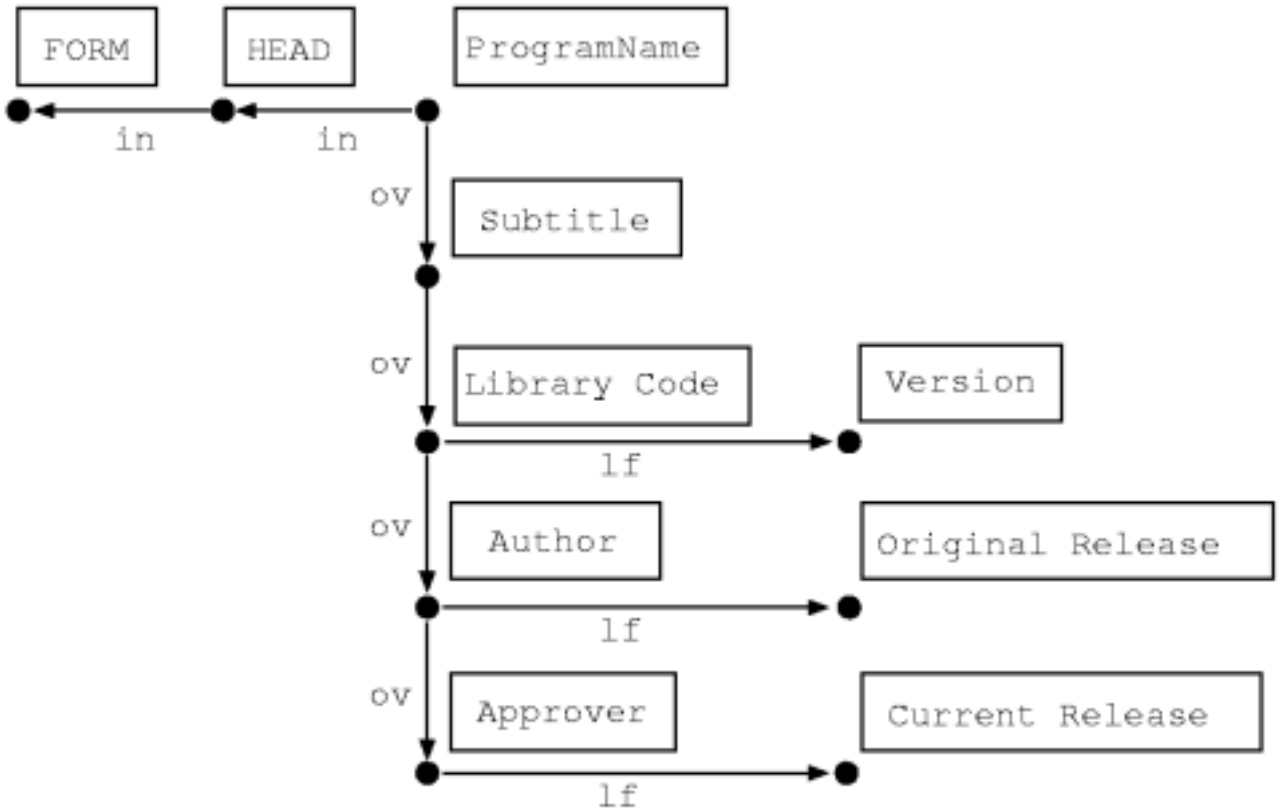
- An Example

Program specification form T1 (This is header part)

ProgramName :	
Subtitle :	
Library Code :	Version :
Author :	Original Release :
Approver :	Current Release :

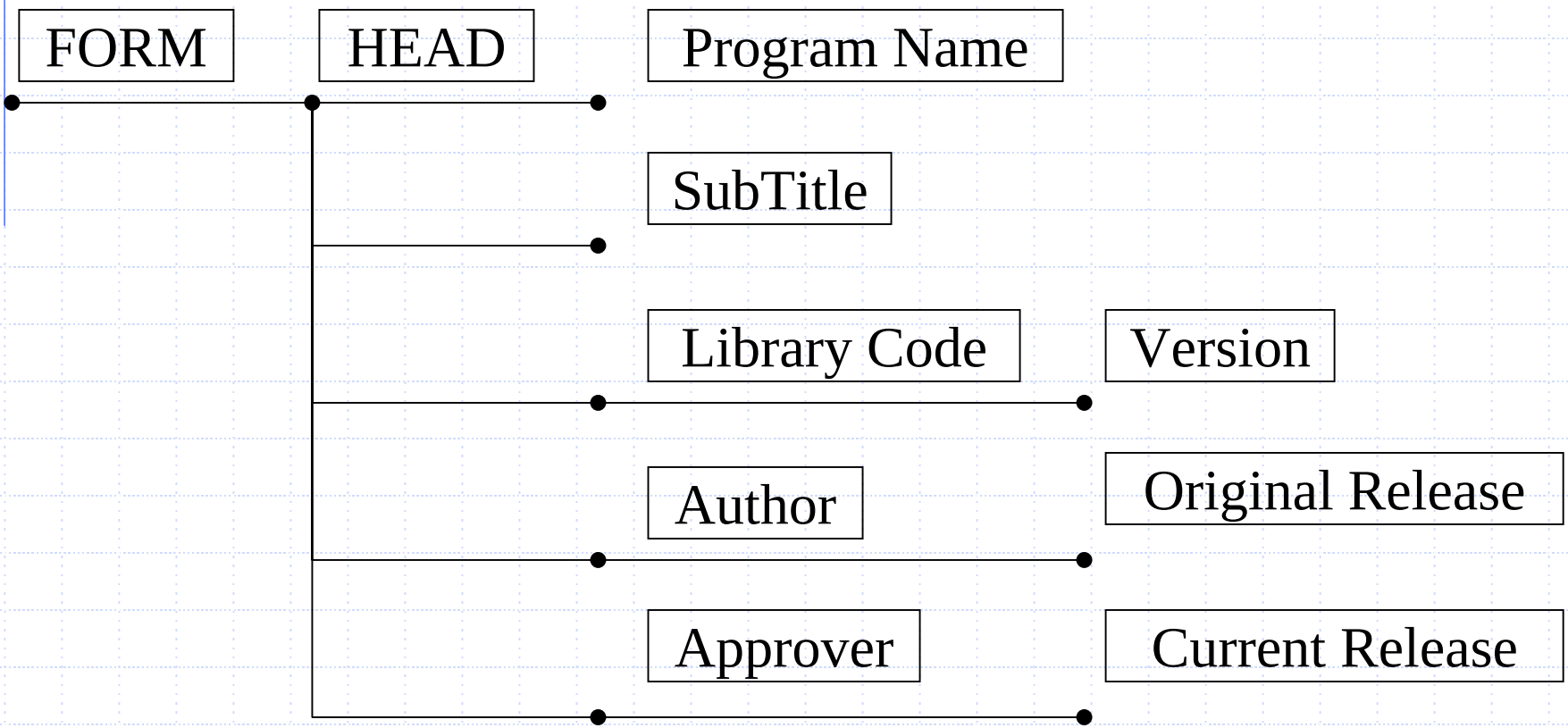
Representation of tabular forms by XML (Contimued)

- Marked graph G1 for T1



Representation of tabular forms by XML (Contimued)

- A tree structure of XML for graph G1



Representation of tabular forms by XML (Contimued)

- XML source for T1

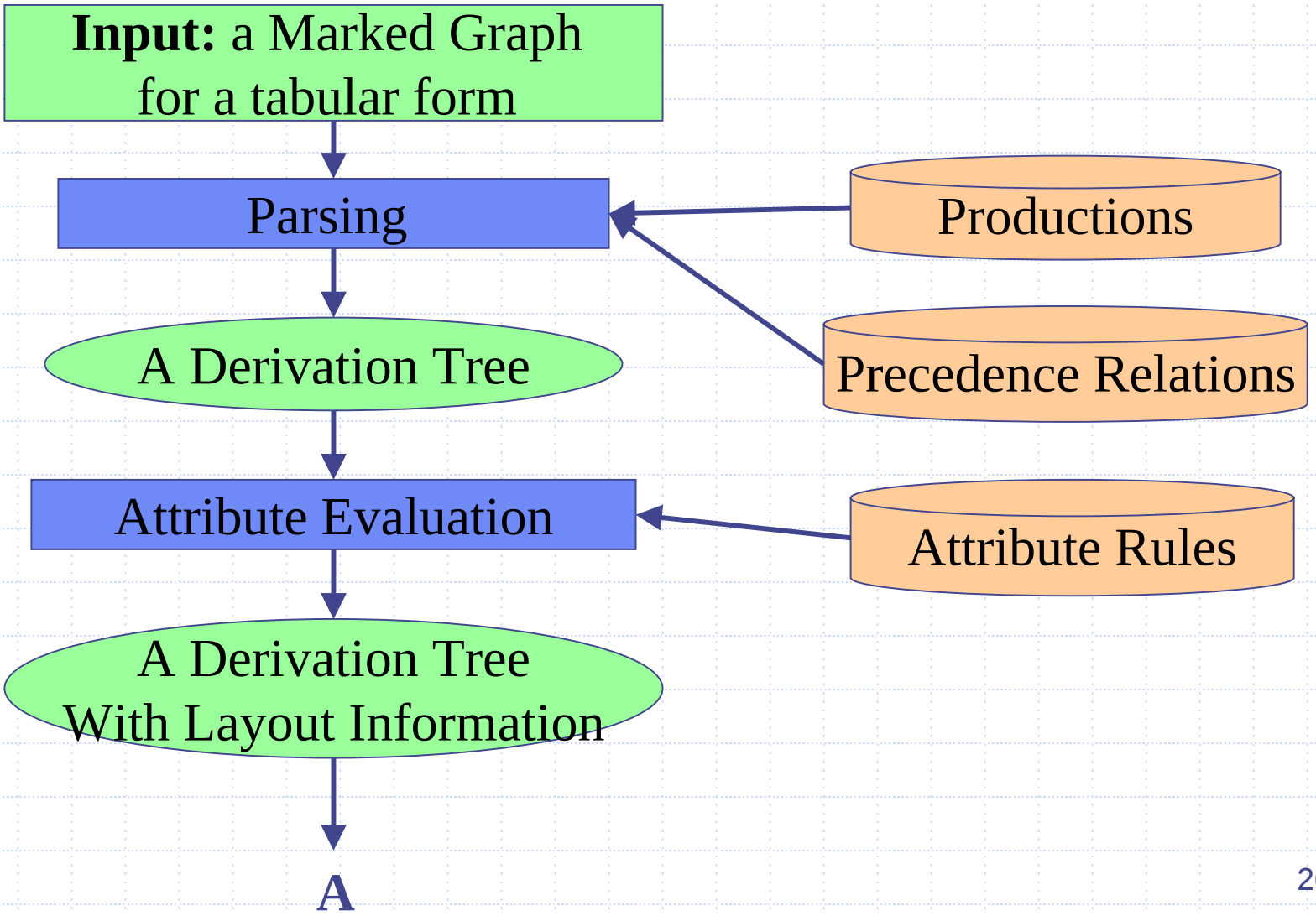
```
<?xml version="1.0" encoding="Shift_JIS"?>
<?xml-stylesheet type="text/xsl" href="Hiform.xsl"?>

<document>
<graph>
    <node id="0" label="FORM" x="0" y="0" width="200" height="810">
        <node id="1" label="HEAD" x="0" y="0" width="200" height="150">
            <node id="2" label="Program Name" width="200" height="30"></node>
            <node id="3" label="Subtitle" width="200" height="30"></node>
            <node id="4" label="Library Code" width="100" height="30">
                <node id="5" label="Version" width="100" height="30"></node>
            </node>
            <node id="6" label="Author" width="100" height="30">
                <node id="7" label="Original Release" width="100" height="30"></node>
            </node>
            <node id="8" label="Approver" width="100" height="30">
                <node id="9" label="Current Release" width="100" height="30"></node>
            </node>
        </node>
    </node>
</graph>
</document>
```

Representation of tabular forms by XML (Contimued)

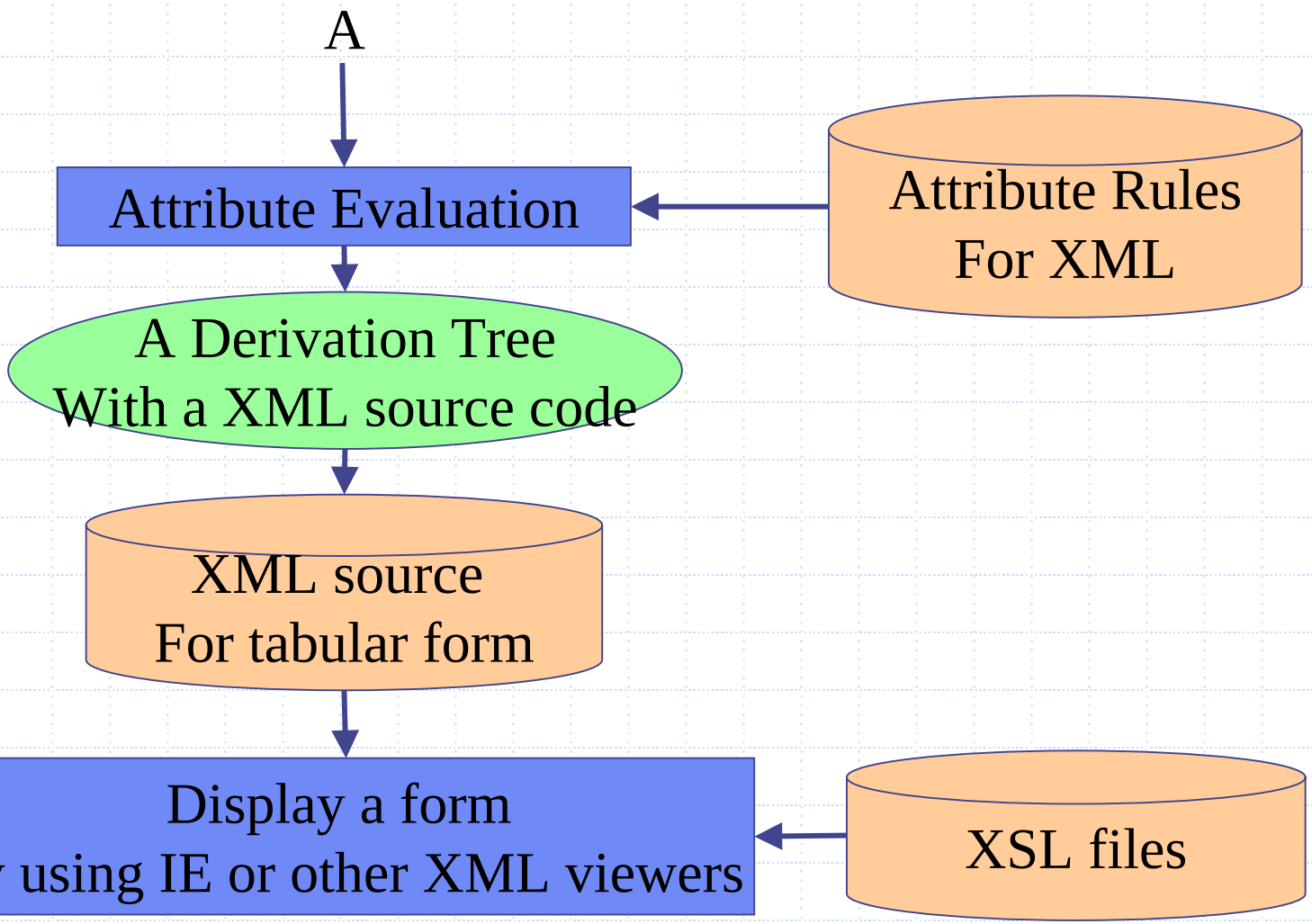
- XML source for displaying T1

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl" xml:lang="ja">

<xsl:template match="/">
:
:
<xsl:template match="node[@label='HEAD']">
<table border="1">
<xsl:for-each test="node">
<tr>
        <td>

                <xsl:if test="@width[.= 100]">
                        <xsl:attribute name="width">
                        400
                        </xsl:attribute>
                </xsl:if>

                <xsl:if test="@width[.= 200]">
                        <xsl:attribute name="colspan">
                        2
                        </xsl:attribute>
                </xsl:if>
:
```

Process flow for generating and viewing XML files

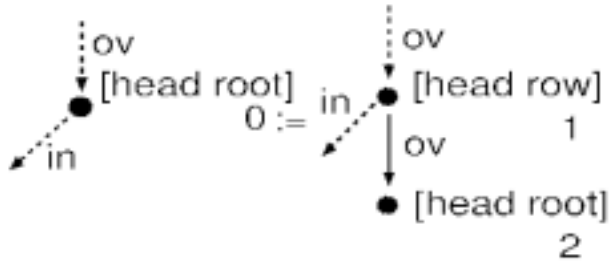
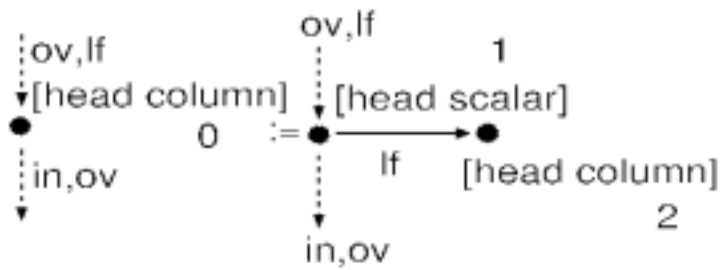


Process flow for generating and viewing XML files (Continued)



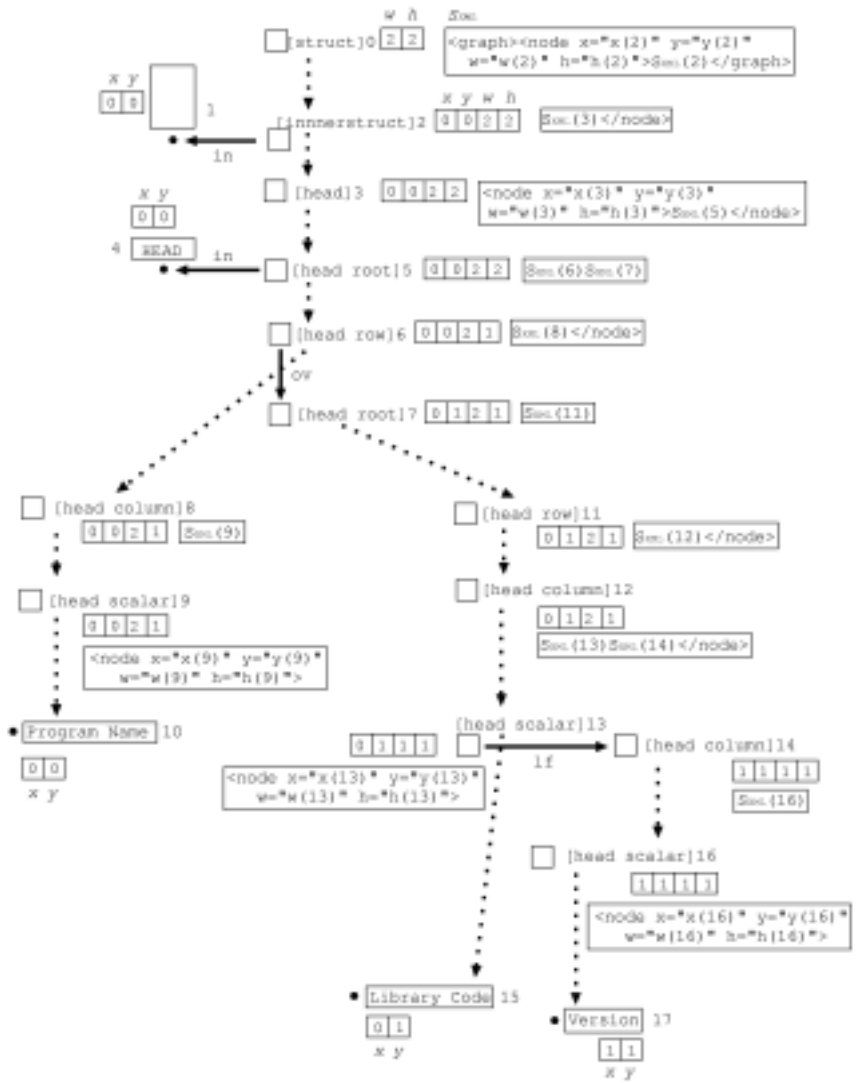
Definitions of attribute rules for generating XML source files

- Attribute S_{XML} for generating XML.
- S_{XML} is defined by concatenation operator " · ".

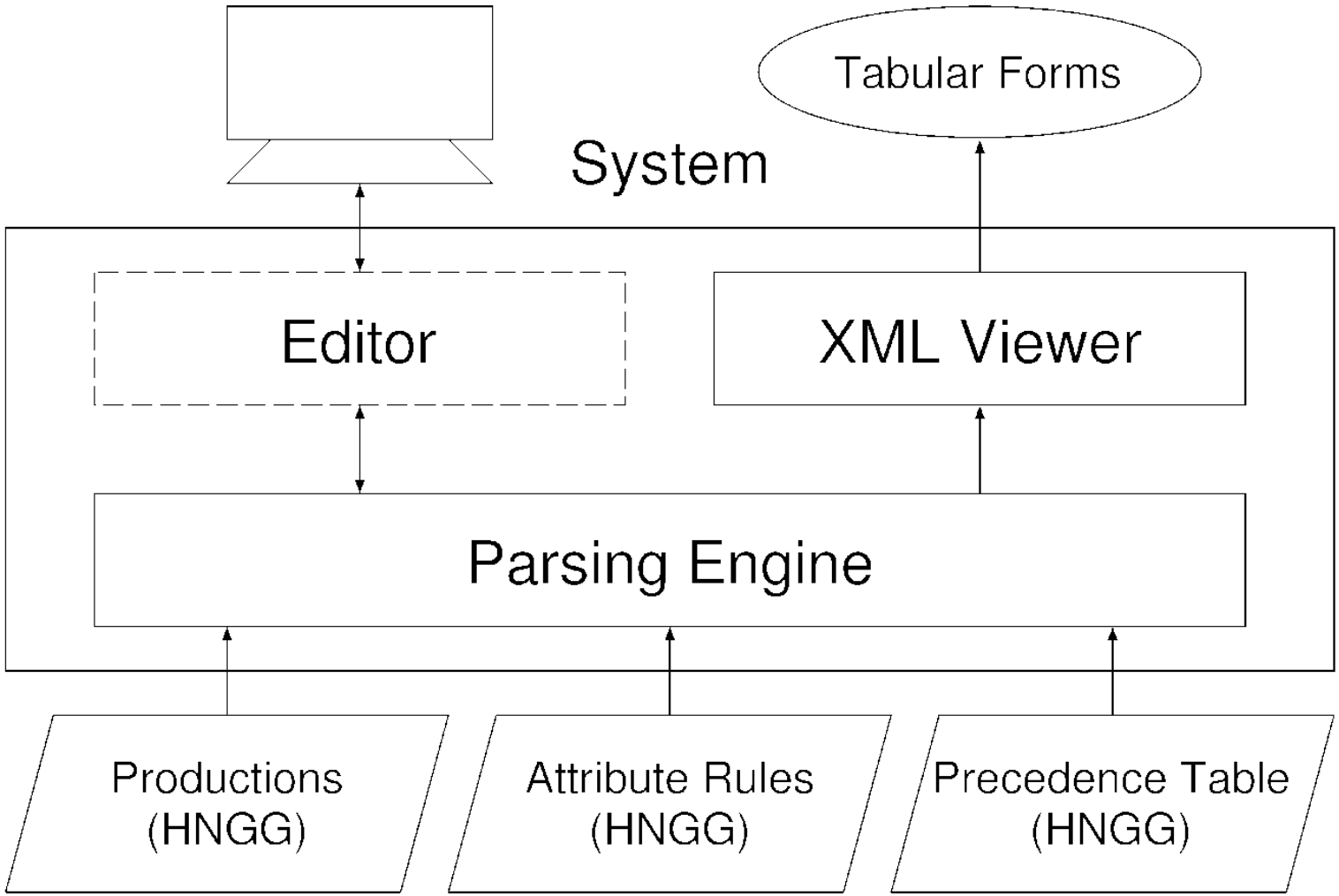
H2		x(1) = x(0) x(2) = x(0) y(1) = y(0) y(2) = y(0) + height(1) width(0) = max(width(1) , width(2)) height(0) = height(1) + height(2) $S_{XML}(0) = S_{XML}(1) \cdot S_{XML}(2)$
H5		x(1) = x(0) x(2) = x(0) + width(1) y(1) = y(0) y(2) = y(0) width(0) = width(1) + width(2) height(0) = max(height(1) , height(2)) $S_{XML}(0) = S_{XML}(1) \cdot S_{XML}(2)$ ·</node>

Definitions of attribute rules for generating XML source files (Continued)

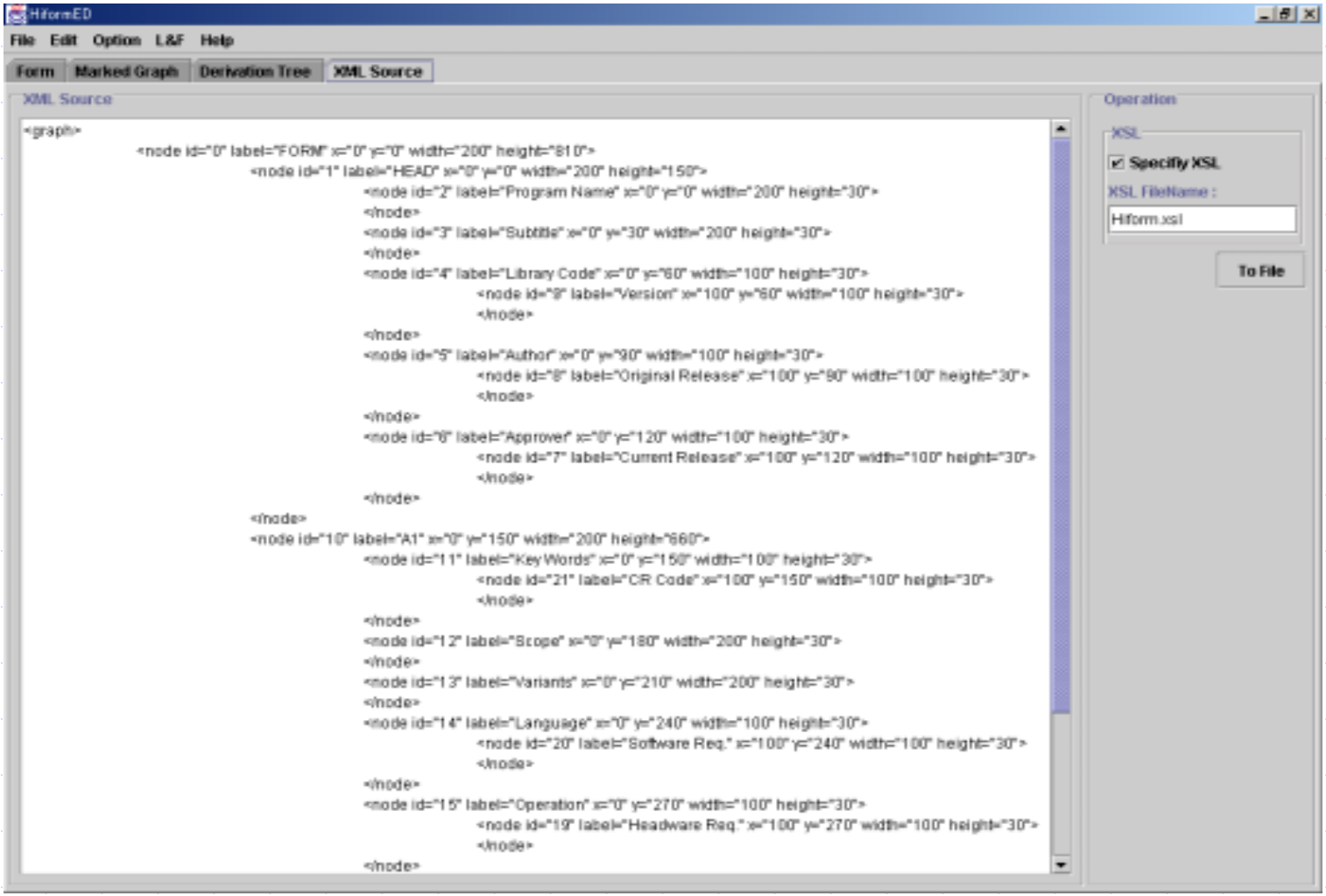
- A derivation tree with S_{XML}



System Structure



Output of XML



Result 1

- View of a XML document by generated from our system on IE 6.

System Structure

Editor	under development
Drawing Engine	3k Java line
Parsing Engine	2k Java line
Productions	280
Attribute rules	1528
Precedence rules	5376

Conclusion

- Constricted XML format for tabular forms
- Constructed XSL style sheet for tabular form
- Extended an attribute graph grammar for generating XML documents
- Considered relations tabular form parser and generation for XML.

Conclusion (Continued)

- This system generates XML sources by attribute evaluation
- The XML viewer is implemented on a tabular form parser and is confirmed whether it is run or not.
- XML sources are displayed on XML viewer such as IE by using XSL files for tabular forms.
- XML parser for tabular form is not developed now.

Definitions of attribute rules for generating XML source files (Continued)

- XMLソースの生成は，前述の属性S_{XML}の属性評価による．
- ボトムアップに評価を進め，根のS_{XML}の属性値が求めるXMLソースとなる．
- 属性S_{XML}を付加した導出木を示す．

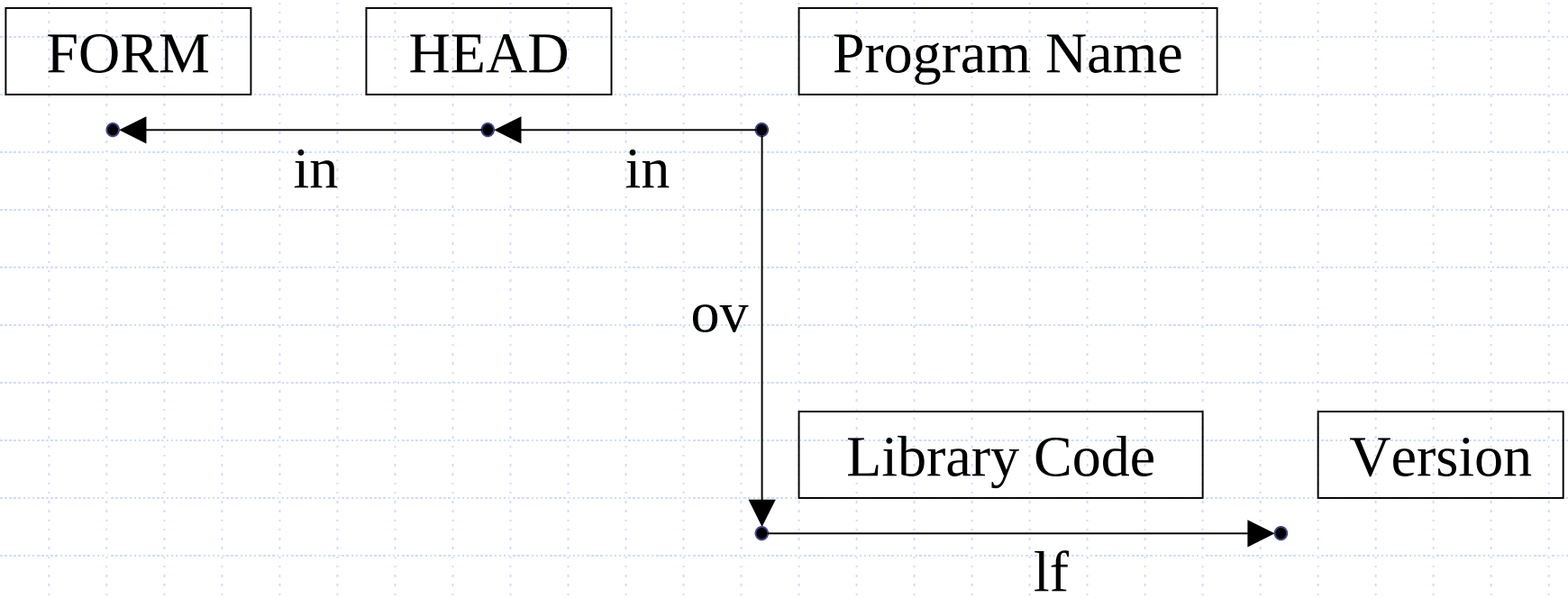
Definitions of attribute rules for generating XML source files (Continued)

- 例

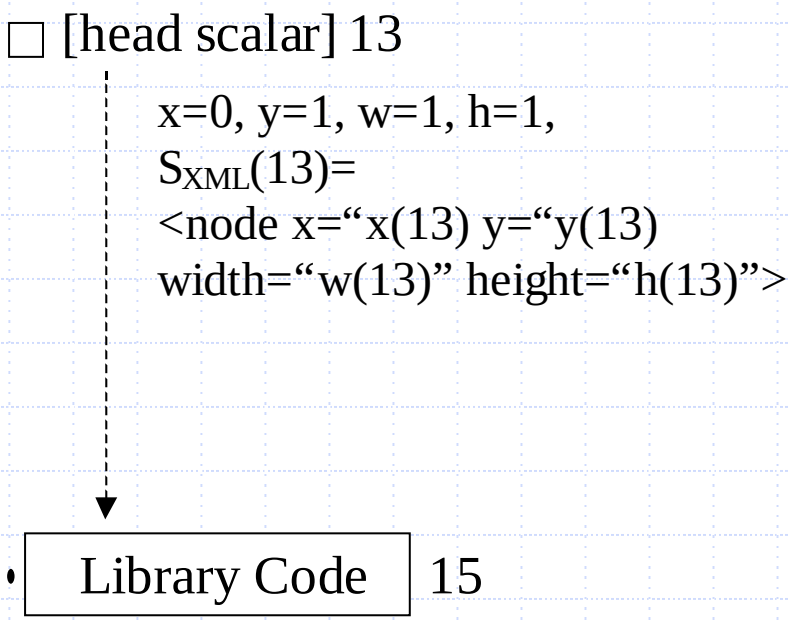
Program Name:	
Library Code:	Version:

Definitions of attribute rules for generating XML source files (Continued)

- 例 (マーク付きグラフ)

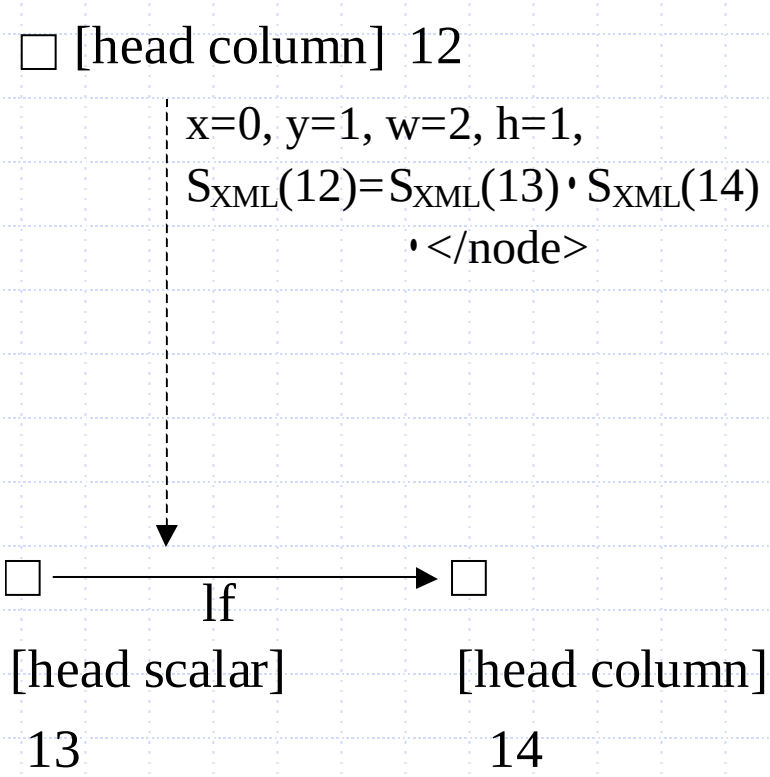


Definitions of attribute rules for generating XML source files (Continued)



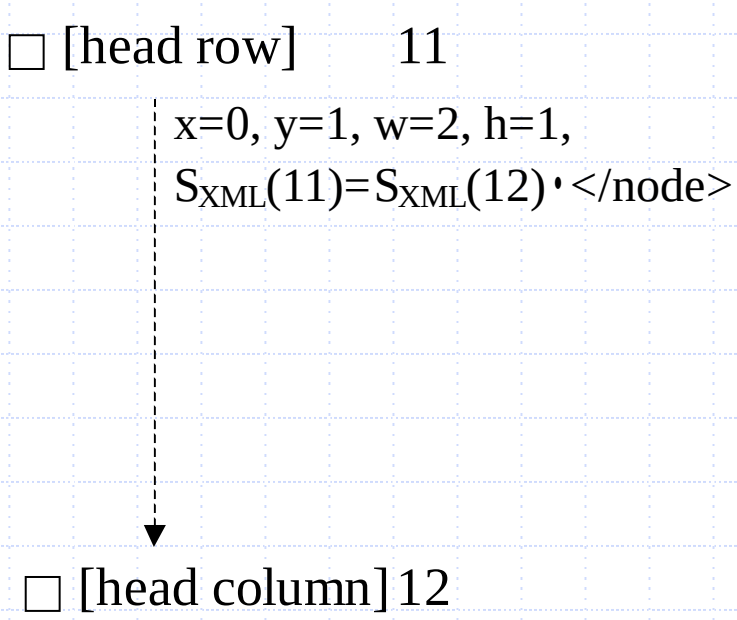
- S_{XML} (13)=
  <node label="Library Code" x="0 y="1" width="1" height="1">

Definitions of attribute rules for generating XML source files (Continued)



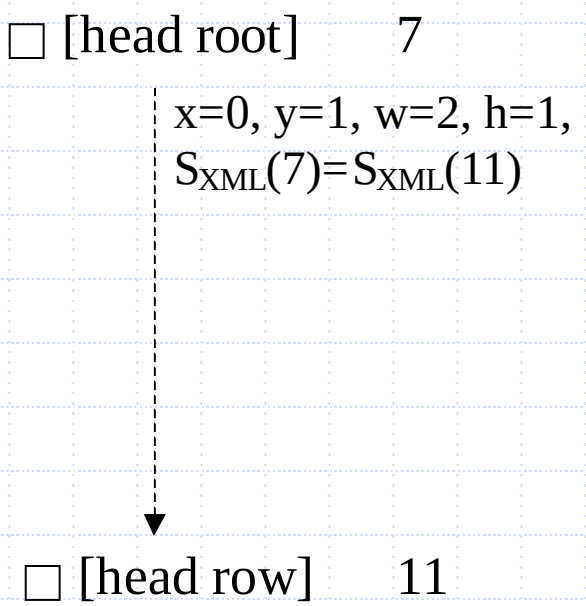
- S_{XML} (12)=
  <node x="0" y="1"
  width="1"
  height="1"> •
  <node x="1"
  y="1" width="1"
  height="1"> •
  </node>

Definitions of attribute rules for generating XML source files (Continued)



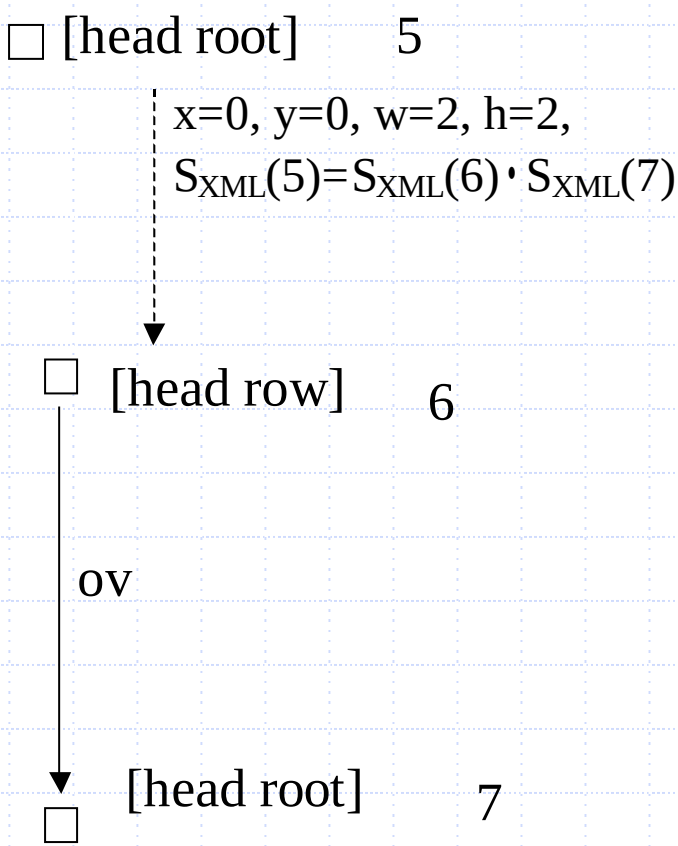
- S_{XML} (11)=
  <node x="0" y="1"
  width="1"
  height="1"> •
  <node x="1"
  y="1" width="1"
  height="1"> •
  </node> •
  </node>

Definitions of attribute rules for generating XML source files (Continued)



- S_{XML} (7)=
  <node x="0" y="1"
  width="1"
  height="1"> •
  <node x="1"
  y="1" width="1"
  height="1"> •
  </node> •
  </node>

Definitions of attribute rules for generating XML source files (Continued)



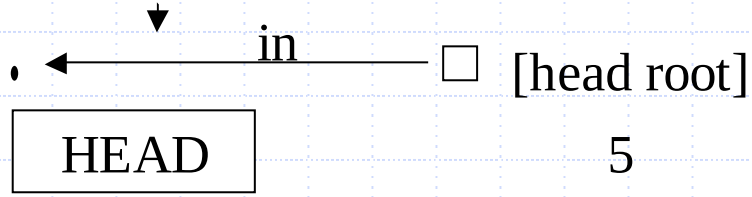
- S_{XML} (5)=

```
<node x="0" y="0"
width="2" height="1">
    <node x="0" y="1"
    width="1"
    height="1"> •
        <node x="1"
        y="1"
        width="1"
        height="1"> •
        </node> •
    </node> •
</node>
```

Definitions of attribute rules for generating XML source files (Continued)

□ [head] 3

x=0, y=0, w=2, h=2,
$S_{XML}(3)$=<node x="x(3) y="y(3)
width="w(3)" height="h
(3)"> ' $S_{XML}(5)$ '
</node>



- S_{XML} (3)=

<node x="0" y="0" width="2"
height="2">
<node x="0" y="0" width="2"
height="1">
<node x="0" y="1"
width="1"
height="1"> •
<node x="1"
y="1"
width="1"
height="1"> •
</node> •
</node> •
</node> •
</node>

Definitions of attribute rules for generating XML source files (Continued)

- [inner struct] 2

  x=0, y=0, w=2, h=2,
  $S_{XML}(2)=S_{XML}(3)$ </node>

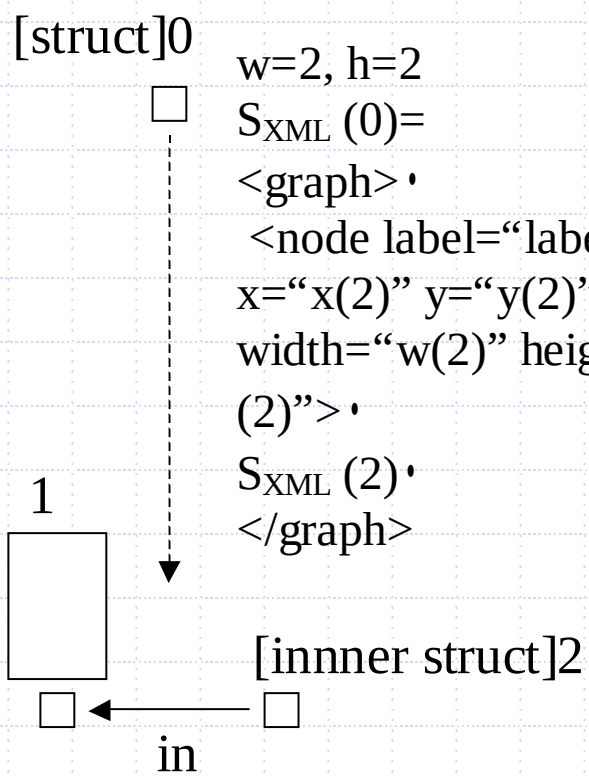
- [head]

  3

- S_{XML} (2)=

```
<node x="0" y="0" width="2"
    height="2">
    <node x="0" y="0" width="2"
    height="1">
        <node x="0" y="1"
        width="1"
        height="1"> •
            <node x="1"
            y="1"
            width="1"
            height="1"> •
            </node> •
        </node> •
    </node> •
</node> •
</node>
```

Definitions of attribute rules for generating XML source files (Continued)



- S_{XML} (0)=

<graph> •

<node label="FORM" x="0" y="0" width="2" height="2"> • </node> •

<node x="0" y="0" width="2" height="2">

<node x="0" y="0" width="2" height="1">

<node x="0" y="1" width="1"height="1"> •

<node x="1" y="1" width="1" height="1"> •

</node> •

</node> •

</node> •

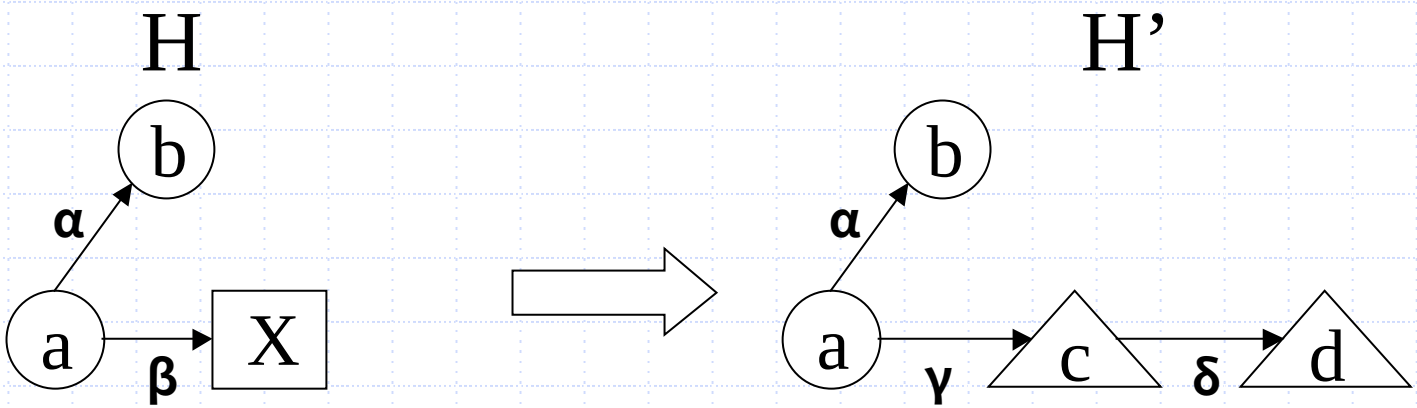
</node>

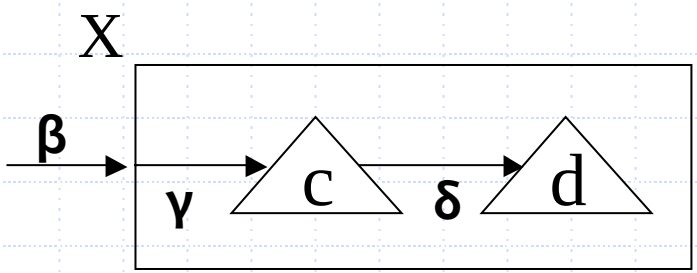
</graph>

Definitions of attribute rules for generating XML source files (Continued)

- 生成したXMLファイルをブラウザで表示.

導出の例





生成規則

Background (Continued)

- Graph grammars for 3 types of diagrams

Graph Grammar	Type of grammar	Diagram
HCGG	Context-free	Hieratical flowchart
HNGG	Context-free and Precedence	Modular Diagram
HTGG	Context-sensitive	Tessellation Tabular Form

Our Parsing System of Tabular Forms

- Graph grammatical definitions for program Specifications
  - IFIP WCC ICS'00
- Parsing methods for tabular forms
  - IASTED AI 2001
- Abstract of Parsing Systems
  - ICSE 2001

グラフ文法によるHiformの定式化

- **定式化されたHiform文法**[IASTED AI 01]
  - HNGG=<G_N, A_N, F_N>
    - G_N=(Σ_N, Δ_N, Γ_N, Ω_N, P_N, S_N)
      - **基底グラフ文法（属性edNCEグラフ文法）**
    - P_N:**生成規則**-280**個**
    - A_N:**属性集合-レイアウト情報及び**XML**ソースを表す**
    - F_N:**属性規則数**-1528**個**
    - HNGG**は順位グラフ文法**
      - 5376**関係の順位**

XML (続き)

- XMLの使用目的は1つに限定することができない
  - Webページの記述
  - システム間メッセージ
  - データ蓄積等
  - 文書やデータの存在するところならばどこにでも適用することが可能

XML (続き)

- XMLによって定義されたマークアップ言語
  - XSL (XMLのスタイルシート言語)
  - MathML (Web上での数学的構造の表現・表示)
  - SMIL (マルチメディアプレゼンテーション用)
  - VML (ベクトルイメージ記述用)
  - SVG (ベクトルイメージ記述用.VMLベース)